PASSED

GroupReport

Akshayaswuroupikka Balasubramanian,yixuan xu

March 1, 2016

Assignment 1: Computations with Metropolis-Hastings

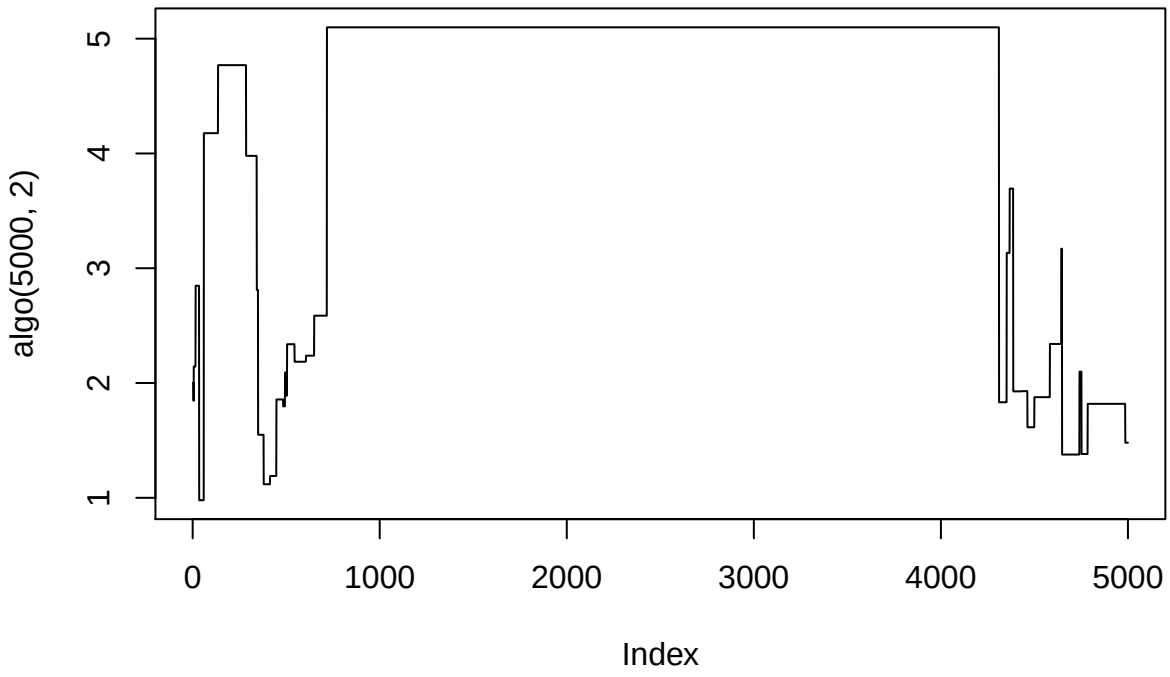
Consider the following probability density function:

$$f(x) \propto x^5 e^{-x}, x > 0$$

You can see that the distribution is known up to some constant of proportionality.

1.1

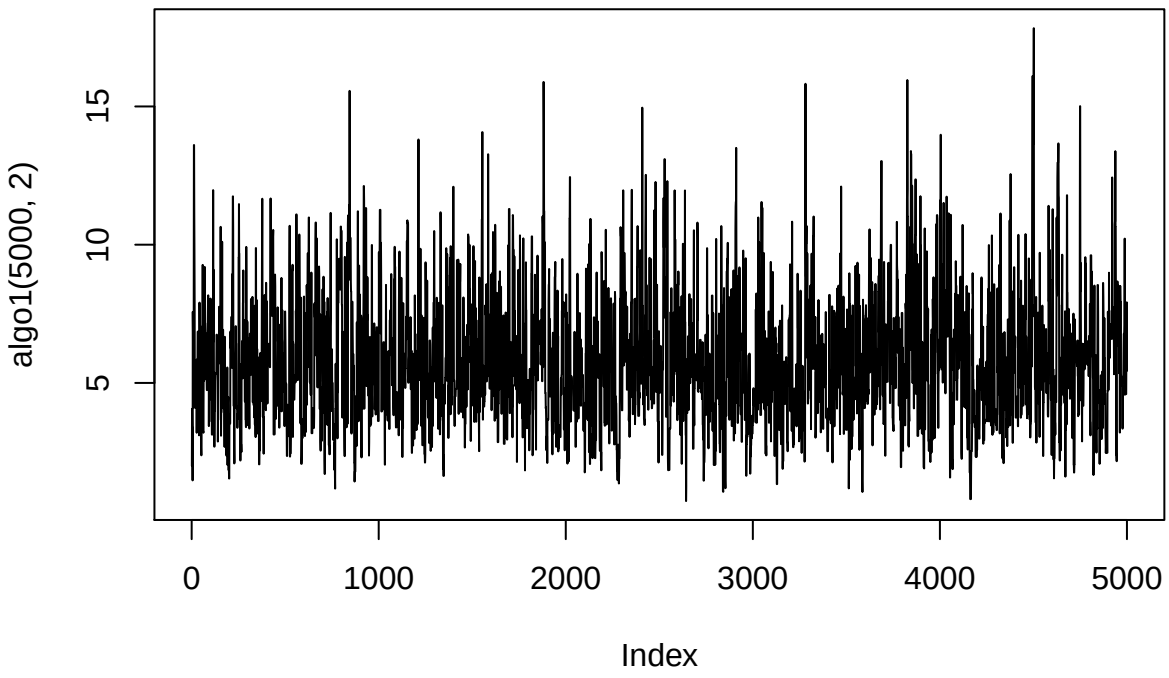
The Metropolis-Hastings algorithm is used to generate samples from the above given distribution by using proposal distribution as log-normal LN(Xt, 1) by taking starting point as 2.



The chain obtained is plotted as a time series plot.The chain is not converged.Sometimes it moves far apart or it stays at a same point.There is no burn in period as far as this is concerned.

1.2

We have to perform step 1 by using chi-square distribution $X_2(floor(X_t + 1))$ as proposal distribution where floor(x) means integer part of x.



1.3

compared to step 1 , step 2 has a plot which converges witha mean approximately equal to 6.Hence,when we are using chi-square distribution as proposal distribution it works better than norm-log distribution as proposal distribution.If a proposal distribution takes small steps, then the acceptance probability will be high, geting a higher rate at which we accept candidate points. When mixing slowly, it will take longer to get to the stationary distribution.

1.4

```
## Potential scale reduction factors:
##
##      Point est. Upper C.I.
## [1,]          1          1
```

The convergence number should vary from 1.0-1.2.In this case we have a convergence number within that range.

1.5

Here we have to estimate $\int_0^\infty Xf(x)dx$ using the samples from steps 1 and 2.

```
## [1] 2.490701
```

```
## [1] 5.956674
```

1.6

The Gamma distribution is given by:

$$\frac{1}{\Gamma(\kappa)\theta^\kappa}x^{\kappa-1}e^{-\frac{x}{\theta}}$$

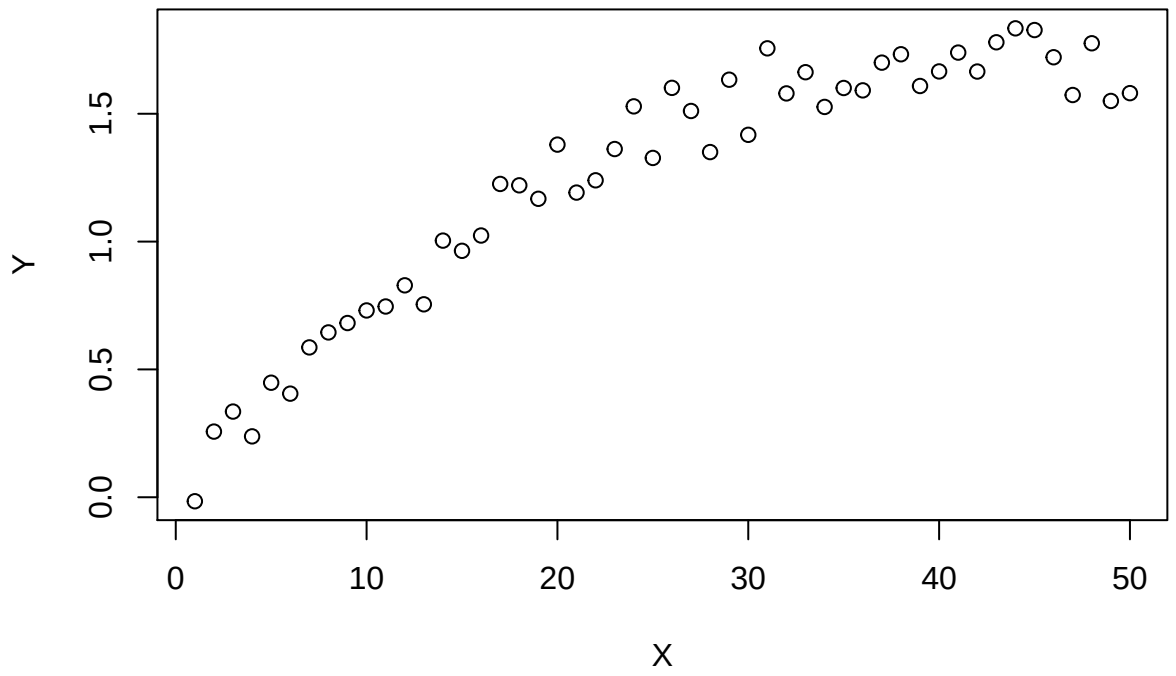
Hence, $\theta = 1$, $\kappa = 6$. The expected value $E[X] = \kappa\theta = 6$ This result is quite close to the result we obtained.

Assignment 2: Gibbs sampling

A concentration of a certain chemical was measured in a water sample, and the result was stored in the data chemical.RData having the following variables: X: day of the measurement Y: measured concentration of the chemical. The instrument used to measure the concentration had certain accuracy; this is why the measurements can be treated as noisy. Your purpose is to restore the expected concentration values.

2.1.

The data is imported to R and the dependence of Y on X is plotted below:



From the plot, quadratic model could be used here.

2.2.

A researcher has decided to use the following (random-walk) Bayesian model (n=number of observations, $\mu = (\mu_1, ..., \mu_n)$ are unknown parameters):

$$Y_i \sim N(\mu_i, var = 0.2), i = 1, ..., n$$

where the prior is

$$p(\mu_1) = 1$$
$$\mu_{i+1} \sim N(\mu_i, 0.2), i = 1, ..., n-1$$

Present the formulas showing the likelihood $p(Y|\mu)$ and the prior $p(\mu)$(hint: a chain rule can be used here $p(\mu) = p(\mu_1)p(\mu_2|\mu_1)p(\mu_3|\mu_2)...p(\mu_n|\mu_{n-1})$)

Likelihood function:

$$P(Y|\mu) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} e\, -\frac{(x-\mu)^2}{2\sigma^2} \sim N(\mu_i, \sigma^2 = 0.2)$$
$$= (0.4\pi)^{-\frac{n}{2}} e^{-\frac{1}{0.4}\sum_{i=1}^{n}(x_i-\mu_i)^2}$$

$p(\mu)$ function:

$$p(\mu_1) = 1$$

$$p(\mu_2|\mu_1) = \frac{1}{\sqrt{0.4\pi}} e^{-\frac{(\mu_2-\mu_1)^2}{0.4}}$$

$$\vdots$$

$$p(\mu_i|\mu_{i-1}) = \frac{1}{\sqrt{0.4\pi}} e^{-\frac{(\mu_i-\mu_{i-1})^2}{0.4}}$$

Since:

$$\begin{aligned} p(\mu) &= p(\mu_1)p(\mu_2|\mu_1)p(\mu_3|\mu_2)...p(\mu_i|\mu_{i-1}) \\ &= (0.4\pi)^{-\frac{n-1}{2}} e^{-\frac{1}{0.4}\sum_{i=1}^{n}(\mu_i-\mu_{i-1})^2} \end{aligned}$$

2.3.

Use the Bayes theorem to get the posterior up to a constant of proportionality, and then find out the distributions for $\mu_i|\mu_{-i}, Y$ where μ_{-i} is a vector containing all μ values except of μ_i

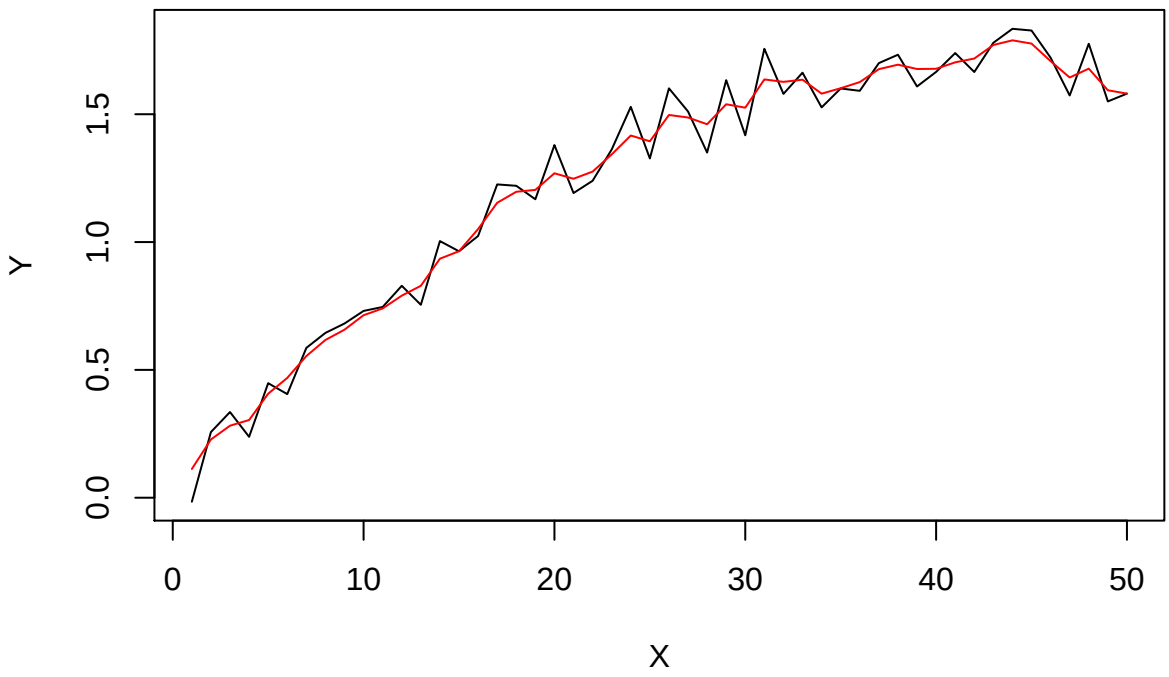
$$p(\mu|Y) = 0.4\pi^{2n-1} e^{-\frac{1}{0.4}(\sum_{i=1}^{n}(y_i-\mu_i)^2)+(\sum_{i=1}^{n}(\mu_i-\mu_{i-1})^2)}$$

$$p_{(\mu_i|\mu_{-i})} \propto e^{-\frac{1}{0.4}(y_i-\mu_i)^2+(\mu_i-\mu_{i-1})^2+(\mu_{i+1}-\mu_{i-1})^2}$$

$$= e^{-\frac{1}{\frac{0.4}{3}}(\mu_i-\frac{(\mu_{i+1}-\mu_{i-1}-y_i)}{3})^2} e^{-\frac{1}{0.4}\left[-\frac{1}{3}(\mu_{i+1}^2+\mu_{i-1}^2+y_i^2)-\frac{8}{9}(y_i\mu_{i+1}+y_i\mu_{i-1}+\mu_{i+1}\mu_i)\right]}$$

Since $e^{-\frac{1}{0.4}\left[-\frac{1}{3}(\mu_{i+1}^2+\mu_{i-1}^2+y_i^2)-\frac{8}{9}(y_i\mu_{i+1}+y_i\mu_{i-1}+\mu_{i+1}\mu_{i-1})\right]}$ is a constant, thus the $\mu_i|\mu_{-i}, Y$ distribution is $N(\frac{y_i+\mu_{i+1}+\mu_{i-1}}{3}, \sqrt{\frac{1}{15}})$. For the first term, the distribution is $N(\frac{\mu_2+y_1}{2}, \sqrt{0.1})$ For the last term, the distribution is $N(\frac{\mu_{49}+y_{50}}{2}, \sqrt{0.1})$

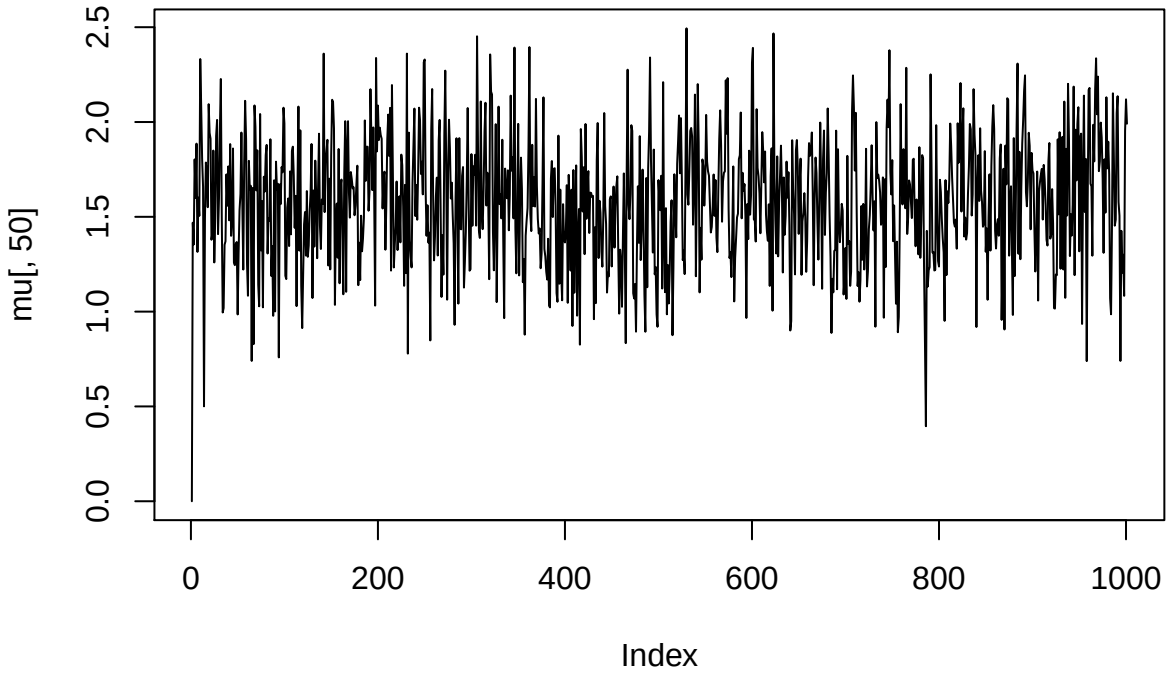
2.4



From the plot, it is easy to see that most of the noise has been removed, the line from the μ has less scale than the line from Y. And the expected value of μ has catched the true underlying dependence, because the line passes through almost all of the points.

2.5

A trace plot for μ_{50} is given below:



There is a burn-in period of 2% of the whole points from the start point. And this plot shows the chain is convergence.

contribution

The first part of the assignment is contributed by akshaya and the second part by Yixuan xu.

Appendix - R-code

```
## ----echo=FALSE----------------------------------------------------------
dist<-function(x){
  a=x^5*exp(-x)
  return(a)
}
algo<-function(n,start){
  x <- NULL
  if(is.null(start)){
    x[1] <- runif(1,0,1)
  }else{
  x[1]<- start
  }
  for(i in 2 : n){
```

```
    y<- rlnorm(1, x[i-1], 1)
    u <- runif(1,0,1)
    frac<- dist(y)*dlnorm(x[i-1],y,1)/(dist(x)*dlnorm(y,x[i-1],1))
    alp=min(1,frac)
    if(u <= alp){
      x[i] <- y
    }else{
      x[i] <- x[i-1]
    }
  }
  return(x)
}

plot(algo(5000,2),type='l')


## ---- echo=FALSE----------------------------------------------------------
algo1<-function(n,start){
  x <- NULL
  if(is.null(start)){
    x[1] <- runif(1,0,1)
  }else{
  x[1]<- start
  }
  for(i in 2:n){
    y<- rchisq(1, floor(x[i-1]+ 1))
    u <- runif(1,0,1)
    frac<-dist(y)*dchisq(x[i-1],floor(y+1))/(dist(x[i-1])*dchisq(y,floor(x[i-1]+1)))
    alp=min(1,frac)
    if(u <= alp){
      x[i] <- y
    }else{
      x[i] <- x[i-1]
    }
  }
  return(x)
}

plot(algo1(5000,2),type='l')

## ----echo=FALSE-----------------------------------------------------------
library(coda)
mc<-matrix(NA,5000,10)
for(i in 1:10){
  mc[,i] <-algo1(5000,i)
}
mc1=mcmc.list()
for(i in 1:10){
  mc1[[i]] = as.mcmc(mc[,i])
}
gelman.diag(mc1)

## ----echo=FALSE-----------------------------------------------------------
```

```
int_LN<- mean(algo(5000,2))
int_LN
int_chisq<-mean(algo1(5000,2))
int_chisq


## ---- echo=FALSE----------------------------------------------------------
load("chemical.RData")
plot(X,Y)

## ---- echo=FALSE----------------------------------------------------------
mu <- matrix(NA, nrow=1001, ncol=50)
mu[1,] <- 0
for(i in 2 : 1001){
  for(j in 1 : 50){
    if(j == 1){
      mu[i,j]<- rnorm(1,(mu[(i-1),(j+1)]+Y[j])/2, sqrt(0.1))
    }else if(j == 50){
      mu[i,j]<- rnorm(1,(mu[i,(j-1)]+Y[j])/2, sqrt(0.1))
    }else{
      mu[i,j]<- rnorm(1,(mu[i,(j-1)]+mu[(i-1),(j+1)]+Y[j])/3,sqrt(1/15))
    }
  }
}
mumatrix <- mu[2:1001,]
mu1 <- colMeans(mumatrix)

plot(y=Y,x=X, type = "l")
points(mu1, col="red", type = "l")

## ---- echo=FALSE----------------------------------------------------------
plot(mu[,50],type="l")

## ----code=readLines(knitr::purl("GroupReport.Rmd", documentation = 1)), eval = FALSE----
## NA
```